

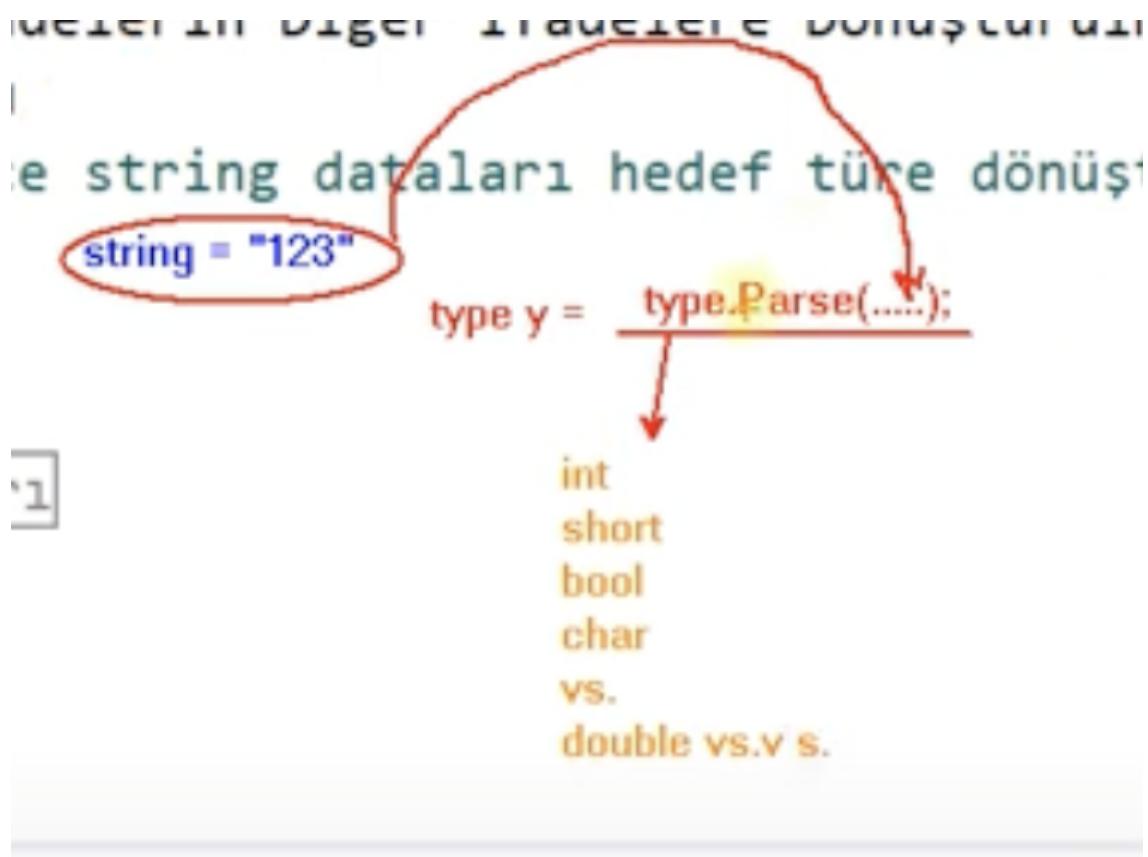
TÜR DÖNÜŞÜMLERİ

Yazılım sürecinde elimizdeki değerlerin türlerini değiştirmek durumunda kalabiliriz. Bunun sebebi ise; Öğrneğin, int bir değeri string şekilde tutup aritmatik bir işlem yapmak istediğimizde tür dönüşümü kullanırız.

Tür dönüşümlerinde amaç türünü değiştirmektir. Elimizdeki veriyi uygun bir tür dönüştürmeliyiz. Uygun olmayan bir tür dönüştürmeye çalışırsak bu mümkün olmayacağından emin olmalıyız. Hata verecektir. Ahmet stringini int e çevirmek gibi.

▼ Metinsel ifadelerin diğer turlere dönüştürülmesi

- PARSE fonksiyonu: Sadece string dataları hedef türde dönüştürürken kullanılır.



- CONVERT fonksiyonu: Sadece string değil, bütün tür dönüşümlerini yapabilir.

```
bir hedef tür belirlenmelidir.  
Parse Metodu  
#region Convert Fonksiyonları  
Örnek 1  
#region Örnek 2  
string x = "3,14";  
double d = Convert.ToDouble(x);  
Console.WriteLine(d * 5);  
endregion  
#endregion
```

▼ Diğer ifadelerin string'e dönüştürülmesi

Tüm veriler stringe dönüştürülebilir, Convert ya da ToString fonksiyonu ile.

```
Convert Fonksiyonu  
#region ToString Fonksiyonu  
//ToString fonksiyonu tüümümm ama tüm türlerde mevcuttur.  
float f = 35;  
string f2 = f.ToString();  
#endregion
```

▼ Sayısal ifadelerin kendi arasındaki tür dönüşümü

- Bir sayısal değer kendinden daha büyük değer aralığına sahip olan diğer türlere dönüştürürken burada herhangi bir işlem yapmamıza gerek kalmayacağı için buna BİLİNÇSİZ tür denir.
- Bir sayısal değer kendinden daha küçük değer aralığına sahip olan diğer türlere dönüştürürken burada hedef türün ilgili veriyi karşılayamam

riskinden dolayı buradaki işlemi bilinçli yapmamız gerekecek. Buna **BİLİNÇLİ** tür dönüşümü denir Veri kaybı olabilir.

Bilinçli tür dönüşümü	Tür	Açıklama ve Bellek Alanı	Max-Min aralığı
	bool	Doğru veya Yanlış (1 Bit)	0-1 (True - False)
	char	Karakterler (16 Bit)	16Bit Unicode
	sbyte	İşaretli Tam Sayı (8 Bit)	-128 ile 127 arası
	byte	İşretsiz Tam Sayı (8 Bit)	0 ile 255 arası
	short	İşaretli Tam Sayı (16 Bit)	-32.768 ile 32.767 arası
	ushort	İşretsiz Tam Sayı (16 Bit)	0 ile 65.535 arası
	int	İşaretli Tam Sayı (32 Bit)	-2.147.483.648 ile 2.147.483.647 arası
	uint	İşretsiz Tam Sayı (32 Bit)	0 ile 4.294.967.295 arası
	long	İşaretli Tam Sayı (64 Bit)	-9.223.372.036.854.775.808 ile 9.223.372.036.854.775.807 arası
	ulong	İşretsiz Tam Sayı (64 Bit)	0 ile 18.446.744.073.709.551.615 arası
	float	Tek Kayan Sayı (32 Bit)	$\pm 1,5 \times 10^{-45}$ ile $\pm 3,4 \times 10^{38}$ arası
	double	Çift Kayan Sayı (64 Bit)	$\pm 5 \times 10^{-324}$ ile $\pm 1,7 \times 10^{308}$ arası
	decimal	Ondalıklı Sayı (128 Bit)	$\pm 1,5 \times 10^{-28}$ ile $\pm 7,9 \times 10^{-28}$ arası

▼ Bilinçsiz tür dönüşümü

- ÖR: int → float'a dönüş

```
#region Bilinçsiz Tür Dönüşümü
int a = 3000;
float f = a; // Bu satırda bir tür dönüşümü söz konusudur.
// Lakin buradaki tür dönüşümü bizim iradeımızla/kararımızla/bilincimizle yaptığımız bir dönüşüm degildir.
#endregion
```

Suanda burada bir tür dönüşümü söz konusudur.
Lakin buradaki tür dönüşümü bizim iradeımızla/kararımızla/bilincimizle yaptığımız bir dönüşüm degildir.

▼ Bilinçli tür dönüşümü

- ÖR: int → short 'a dönüştürürken veri kaybı olabilir. Buna compiler karar vermek istemez, buna biz karar veririz. Bilinçli tür dönüşümü yaparken cast operatörünü kullanırız.

•

```
int x = 3000;
short y = x; // dediğimizde hata verir compiler. Bilemez ve verir.
// -----
// Şu şekilde yaparsak:
```

```
int x = 3000;
short y = short(x); // dersek sorun ortadan kalkar. Veri kaybı
```

//Bilinçili bilinçsiz tür dönüşümüne örnek

```
int a = 3000;
short b = (byte)a; // int olan a'yı byte'a çevirirken bilinçsiz
//byte'a çevirdiğimiz a'yı short olan b'ye atarken bilinçsiz
```

▼ Check bloğu nedir?

Bilinçli tür dönüşümlerinde herhangi bir hata yapmamız durumunda bizi uyarıp hata fırlatan kod bloğudur.

```
checked
{
    // veri kaybı olabileceğini düşündüğümüz kodu buraya yazarız
}
```

▼ uncheck bloğu nedir?

Bilinçli tür dönüşümlerinde herhangi bir şey yapmadan veri kaybı olsa da olmasa da takmaz. Aslında default olan budur tür dönüşümünde.

▼ Özel tür dönüşümleri

▼ bool → sayısal türe dönüştürme

```
bool a = true;
int b = Convert.ToInt32(a); //Console'a sayısal değerini döndürür
//true ise console'a 1 yazdırır, false ise 0.
```

▼ sayısal türün → bool dönüştürme

int i değeri eğer 0 ya da 1 den farklı bir değer alırsa, bize yine true döndürür. Şöyle düşünebiliriz: 0 verilemediği sürece 1 döndürür.

```
int i =1;
int b = Convert.ToBoolean(i); //Console'a mantıksal değer

// 1 ise console'a true yazdırır, 0 ise false.
```

▼ char → sayısal türe dönüştürme (ASCII)

ASCII: Bilgisayardaki her bir karakterin sayısal bir karşılığı vardır. Bu sayısal değerlere ASCII kaynak değerleri denmektedir.

Bir karakteri int cast edersek, ascii kaynak kodunu verecektir

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0 000	NUL	(null)	32	20 040	 	Space		64	40 100	@	R	96	60 140	`	`		
1	1 001	SOH	(start of heading)	33	21 041	!	!	!	65	41 101	A	A	97	61 141	a	a		
2	2 002	STX	(start of text)	34	22 042	"	"	"	66	42 102	B	B	98	62 142	b	b		
3	3 003	ETX	(end of text)	35	23 043	#	#	#	67	43 103	C	C	99	63 143	c	c		
4	4 004	EOT	(end of transmission)	36	24 044	$	\$	\$	68	44 104	D	D	100	64 144	d	d		
5	5 005	ENQ	(enquiry)	37	25 045	%	%	%	69	45 105	E	E	101	65 145	e	e		
6	6 006	ACK	(acknowledge)	38	26 046	&	&	&	70	46 106	F	F	102	66 146	f	f		
7	7 007	BEL	(bell)	39	27 047	'	'	'	71	47 107	G	G	103	67 147	g	g		
8	8 010	BS	(backspace)	40	28 050	(((72	48 110	H	H	104	68 150	h	h		
9	9 011	TAB	(horizontal tab)	41	29 051)))	73	49 111	I	I	105	69 151	i	i		
10	A 012	LF	(NL line feed, new line)	42	2A 052	*	*	*	74	4A 112	J	J	106	6A 152	j	j		
11	B 013	VT	(vertical tab)	43	2B 053	+	+	+	75	4B 113	K	K	107	6B 153	k	k		
12	C 014	FF	(NP form feed, new page)	44	2C 054	,	,	,	76	4C 114	L	L	108	6C 154	l	l		
13	D 015	CR	(carriage return)	45	2D 055	-	-	-	77	4D 115	M	M	109	6D 155	m	m		
14	E 016	SO	(shift out)	46	2E 056	.	.	.	78	4E 116	N	N	110	6E 156	n	n		
15	F 017	SI	(shift in)	47	2F 057	/	/	/	79	4F 117	O	O	111	6F 157	o	o		
16	10 020	DLE	(data link escape)	48	30 060	0	0	0	80	50 120	P	P	112	70 160	p	p		
17	11 021	DC1	(device control 1)	49	31 061	1	1	1	81	51 121	Q	Q	113	71 161	q	q		
18	12 022	DC2	(device control 2)	50	32 062	2	2	2	82	52 122	R	R	114	72 162	r	r		
19	13 023	DC3	(device control 3)	51	33 063	3	3	3	83	53 123	S	S	115	73 163	s	s		
20	14 024	DC4	(device control 4)	52	34 064	4	4	4	84	54 124	T	T	116	74 164	t	t		
21	15 025	NAK	(negative acknowledge)	53	35 065	5	5	5	85	55 125	U	U	117	75 165	u	u		
22	16 026	SYN	(synchronous idle)	54	36 066	6	6	6	86	56 126	V	V	118	76 166	v	v		
23	17 027	ETB	(end of trans. block)	55	37 067	7	7	7	87	57 127	W	W	119	77 167	w	w		
24	18 030	CAN	(cancel)	56	38 070	8	8	8	88	58 130	X	X	120	78 170	x	x		
25	19 031	EM	(end of medium)	57	39 071	9	9	9	89	59 131	Y	Y	121	79 171	y	y		
26	1A 032	SUB	(substitute)	58	3A 072	:	:	:	90	5A 132	Z	Z	122	7A 172	z	z		
27	1B 033	ESC	(escape)	59	3B 073	;	;	;	91	5B 133	[[123	7B 173	{	[
28	1C 034	FS	(file separator)	60	3C 074	<	<	<	92	5C 134	\	\	124	7C 174	|	\		
29	1D 035	GS	(group separator)	61	3D 075	=	=	=	93	5D 135]]	125	7D 175	}]		
30	1E 036	RS	(record separator)	62	3E 076	>	>	>	94	5E 136	^	^	126	7E 176	~	^		
31	1F 037	US	(unit separator)	63	3F 077	?	?	?	95	5F 137	_	_	127	7F 177		_		

▼ sayısal türü → char dönüştürme

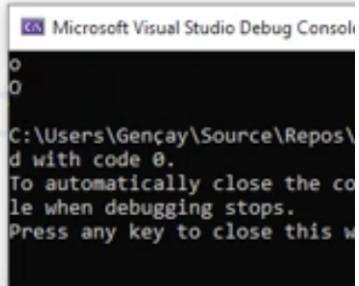
elimizdeki int değerini verdiğimizde ascii kaynağına karşılık gelen char'ı verecektir.

Klavye üzerindeki bütün tuşların ascii değeri vardır.

Tüm tam sayıarda geçerlidir.

```
0 references
static void Main(string[] args)
{
    #region Sayısal Türlerin char Türüne Dönüşürülmesi
    int oascii = 111;
    int Oascii = 79;

    Console.WriteLine((char)oascii);
    Console.WriteLine((char)Oascii);
    #endregion
}
```



Microsoft Visual Studio Debug Console

o
O

C:\Users\Gencay\Source\Repos\
d with code 0.
To automatically close the co
le when debugging stops.
Press any key to close this w